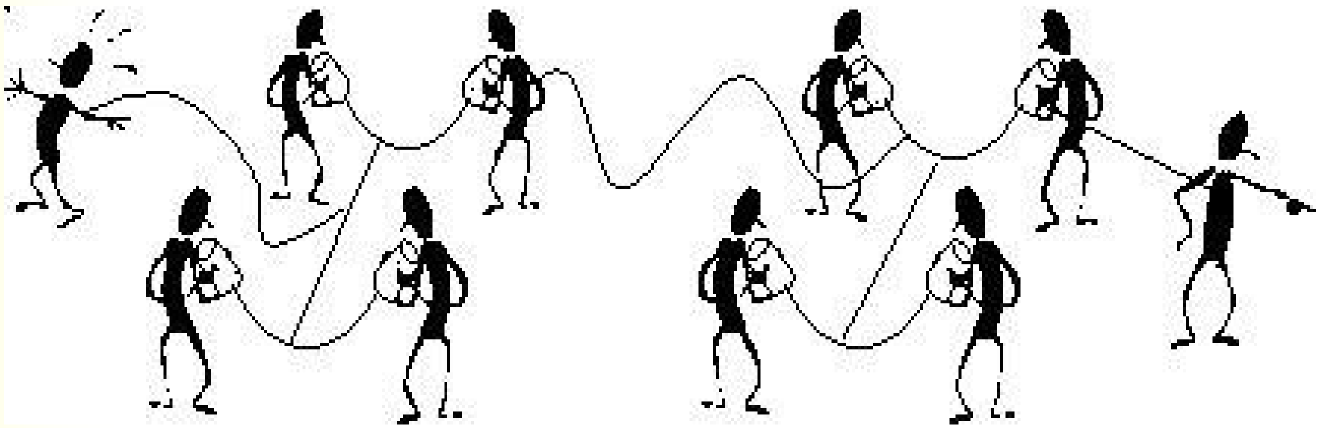
Two Formal Views of Authenticated Group Diffie-Hellman Key Exchange



E. Bresson (ENS), O. Chevassut (LBL, UCL), O. Pereira (UCL)
D. Pointcheval (ENS), J.-J. Quisquater (UCL)

Outline of the Talk

- Introduction to the problem
- A logical approach
- A computational approach
- Discussion and Conclusions…

Key Exchange

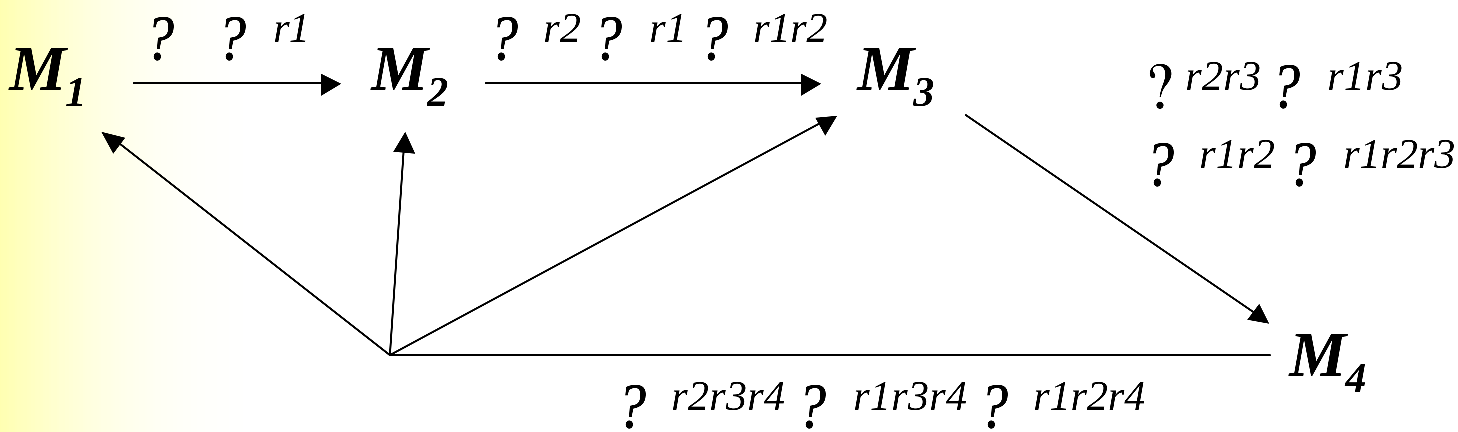
- It is one of the fundamental problems in computer security
- One of the most widespread solutions:
  The Diffie-Hellman protocol

$$A \underset{?^{y}}{\overset{?^{x}}{\rightleftarrows}} B \qquad \text{the secret key is } ?^{xy}$$

- We consider extensions of this protocol enabling a pool of principals to share a key
- The constitution of this pool can dynamically change
- We require authentication properties

Group D.-H. Key Exchange

A possible extension… (Steiner, Tsudik, Waidner, 1996)



a is a generator of a publicly known group

r_i are random fresh contributions

The secret key is $?^{r1r2r3r4}$

Group D.-H. Key Exchange

Benefits:

- Hardness of the Group Decisional Diffie-Hellman (*G-DDH*) problem is implied by the one of the *DDH* problem (Steiner, Tsudik, Waidner, 1996)
- No need of a centralized server
- This scheme allows to dynamically change the group constitution at low-cost…

N.B.: Several other methods for building the key have been proposed (trees, other ways of computing, …)

A Problem remains:

- We need authentication…

Authenticated Key Exchange

- Problem:

$$A \underset{?^{z}}{\overset{?^{x}}{\rightleftarrows}} Z \underset{?^{y}}{\overset{?^{z}}{\rightleftarrows}} B$$

Computes $?^{xz}$ Computes $?^{yz}$

Transformation of the Diffie-Hellman Key Exchange

- We assume that *A* and *B* are sharing a secret K_{AB}

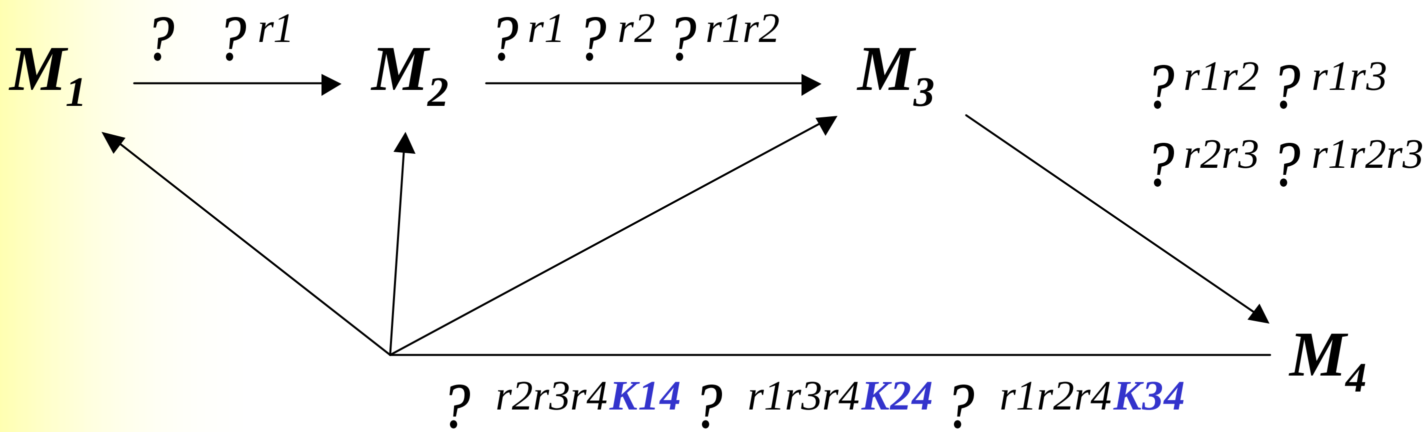
$$A \underset{?^{yKAB}}{\overset{?^{x}}{\rightleftarrows}} B$$

the secret key is $?^{xy}$

We are not able to obtain any key be it computed by *A* or *B*

A-GDH.2 Protocol

- First authenticated group key exchange protocol based on the previous ring scheme (Ateniese, Steiner, Tsudik, 1998)



- K_{ij} is a secret key shared by M_i and M_j
- M_1 computes its key as $?^{\,r1r2r3r4} = (?^{\,r2r3r4K14})^{(r1/K14)}$

Security Properties

- (*Implicit*) *Key Authentication* :
  - Each group member is assured that no party external to the group can obtain (or distinguish) the key he computed
- *Perfect Forward Secrecy* :
  - Compromise of long-term secrets does not imply compromise of past session keys
- *Resistance to Known-Keys Attacks* :
  - Compromise of past session secrets cannot imply compromise of new session keys

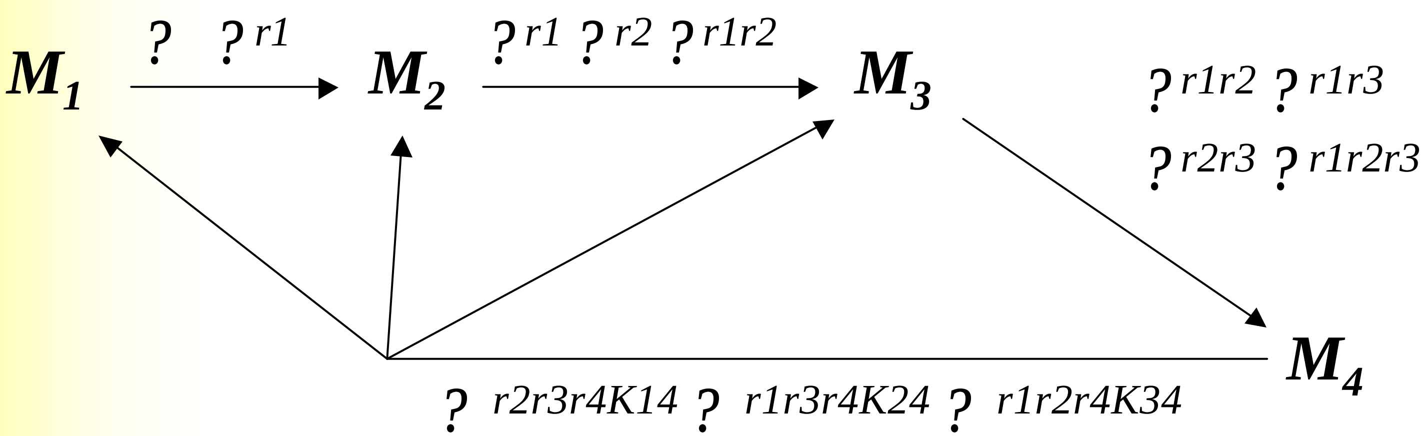
A model for A-GDH Protocols

Computational View	*Logical View*
Random Oracle Paradigm, Standard Model, …	Use of logic, state exploration, nominal calculus, …
Messages considered as strings of bits	Symbolic Representation of Messages
Probabilistic Security Properties	Formal Expression of Security Properties

- We adopted a « logical » (rather than « computational ») point of view

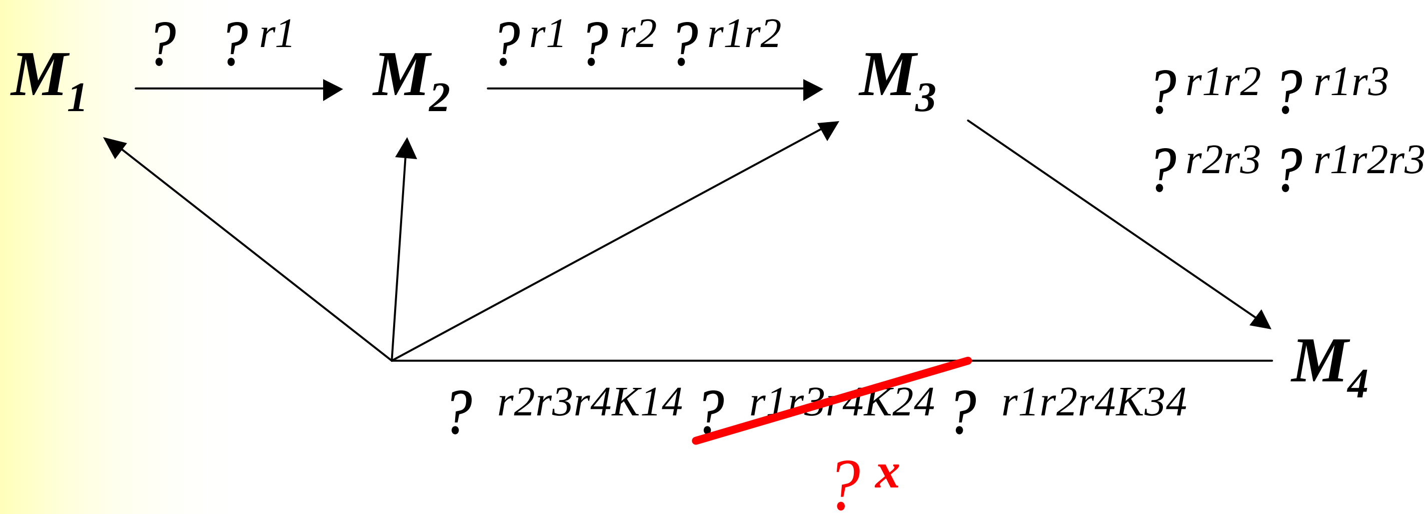
A model for A-GDH Protocols

- Observation:
  - In this family of protocols, the secret key is always computed in the same way:
    M_i receives $?^{x}$ and computes $(?^{x})^{?\, ri\, ?\, Kij}$



A model for A-GDH Protocols

- So, for instance, if an active attacker can obtain (or compute) a pair of elements of the group like $(?^{x}, ?^{x\,r2/K24})$, he can fool M_2:



since M_2 will compute the secret key as $?^{x\,r2/K24}$

Intruder's Knowledge

- How can the intruder obtain such pairs?

1. If he knows ($?^{x}$, $?^{y}$) and z then
   the intruder can compute ($?^{x}$, $?^{yz}$) and ($?^{xz}$, $?^{y}$)

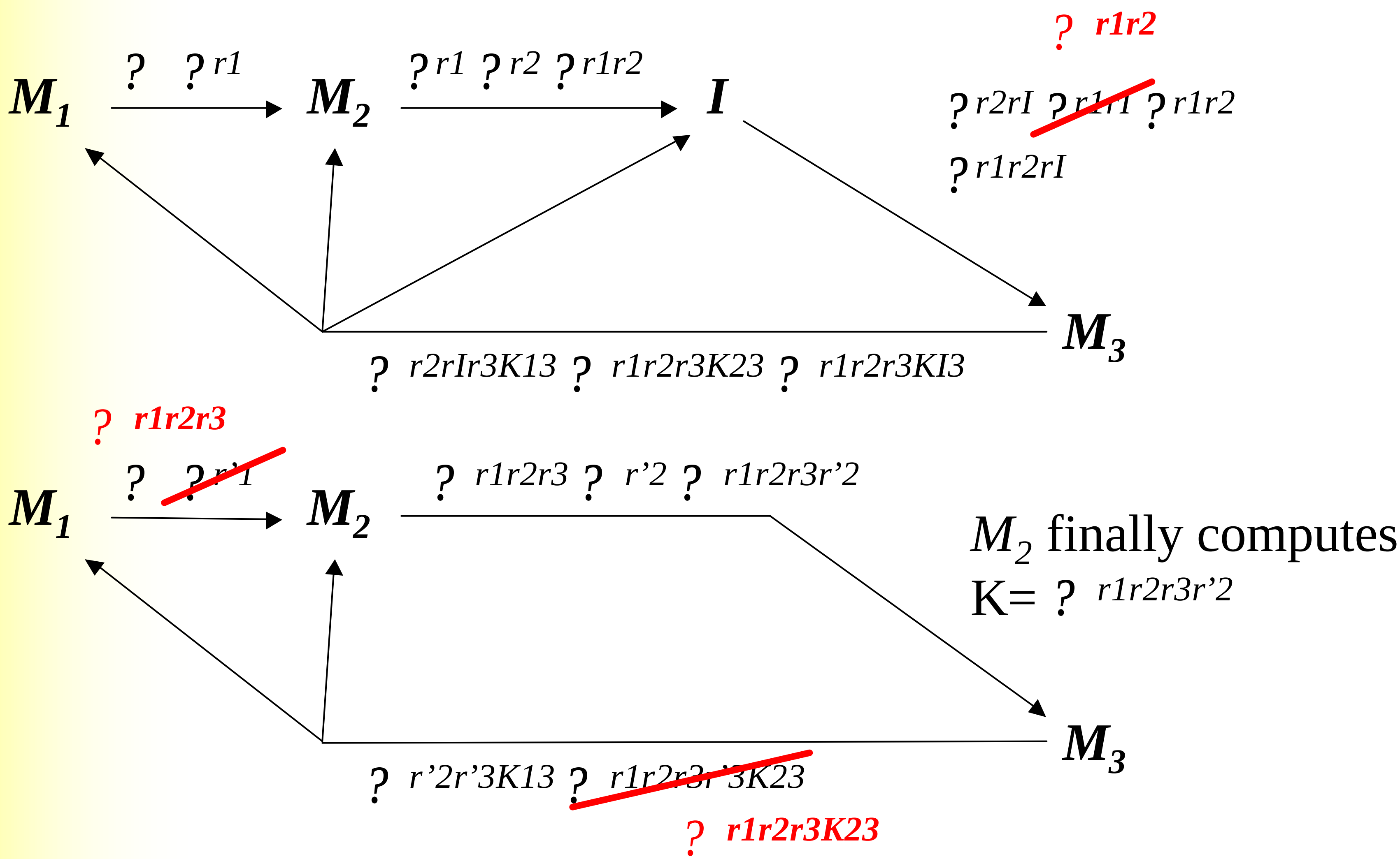
2. If he knows ($?^{x}$, $?^{y}$) and
   if a honest user provides a *service* where
   he transforms $?^{z}$ into $?^{zt}$ then
   the intruder can obtain ($?^{xt}$, $?^{y}$) or ($?^{x}$, $?^{yt}$)

Protocol Analysis

- Having defined our model, we obtained a polynomial algorithm allowing us to check the security of a protocol
  - The verification amounts to solve a linear equation system
- We discovered independent flaws against each security properties in the A-GDH.2 protocol as well as in the SA-GDH.2 protocol
- We also better understood these security properties, that are not simply the transposition of 2-parties properties

Example of Attack

- Against Implicit Key Authentication



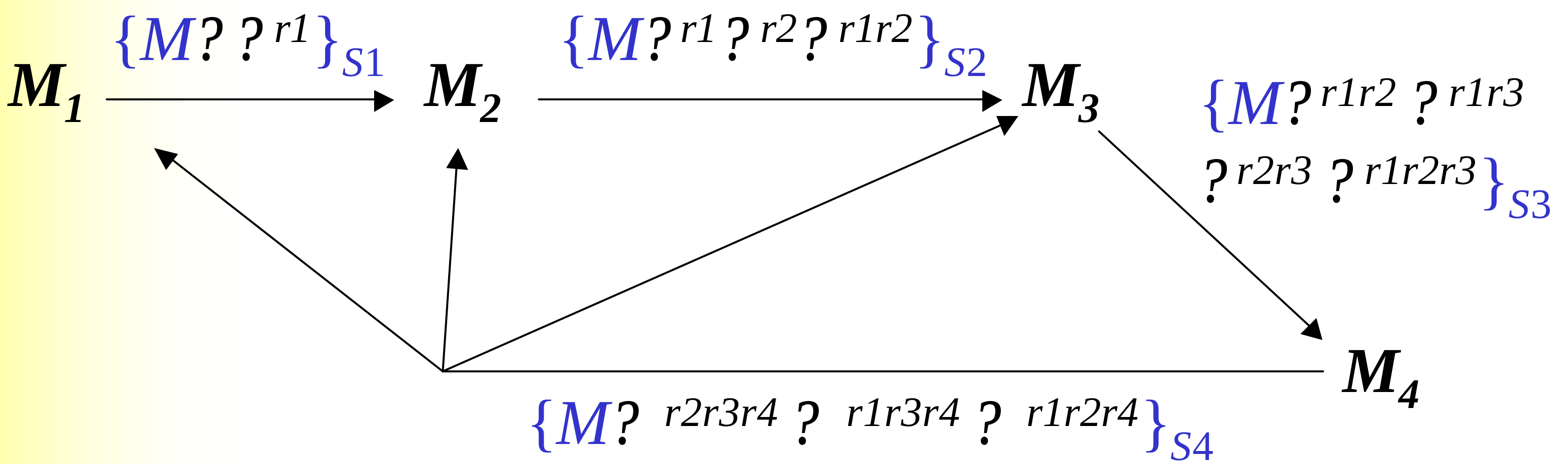
Conclusions

- We defined a logical model for the analysis of a family of protocols
- We discovered several new attacks independently of any computational assumption

- We conjecture that our model could be used to prove that it is impossible to build a protocol using these “constituting blocks” and providing the intended security properties

Another Solution

Obtain Authentication via a Signature Algorithm



$M = M_1\ M_2\ M_3\ M_4$

$\{m\}_{Si}$ is the signature of m through M_i's Long-Lived Key

The key $K = \mathrm{H}(M \| Fl_4 \| ?^{\ r1r2r3r4})$ where H is a universal hash function and Fl_4 is the last flow of the protocol

Another Model

Standard Assumptions:

- Group Decisional Diffie-Hellman
- Multi-Decisional Diffie-Hellman
- Message Authentication Codes (MAC)
- Entropy-smoothing functions

Diffie-Hellman-type Assumptions

- *Group Decisional Diffie-Hellman Problem*
  Given $?^{a}$, $?^{b}$, $?^{c}$, $?^{ab}$, $?^{ac}$, $?^{bc}$,
  Distinguish $?^{abc}$ from a random value $?^{r}$.

- *Multi-Decisional Diffie-Hellman Problem*
  Given $?^{a}$, $?^{b}$, $?^{c}$,
  Distinguish $?^{ab}$, $?^{ac}$, $?^{bc}$ from three random values $?^{r}$, $?^{s}$, $?^{t}$

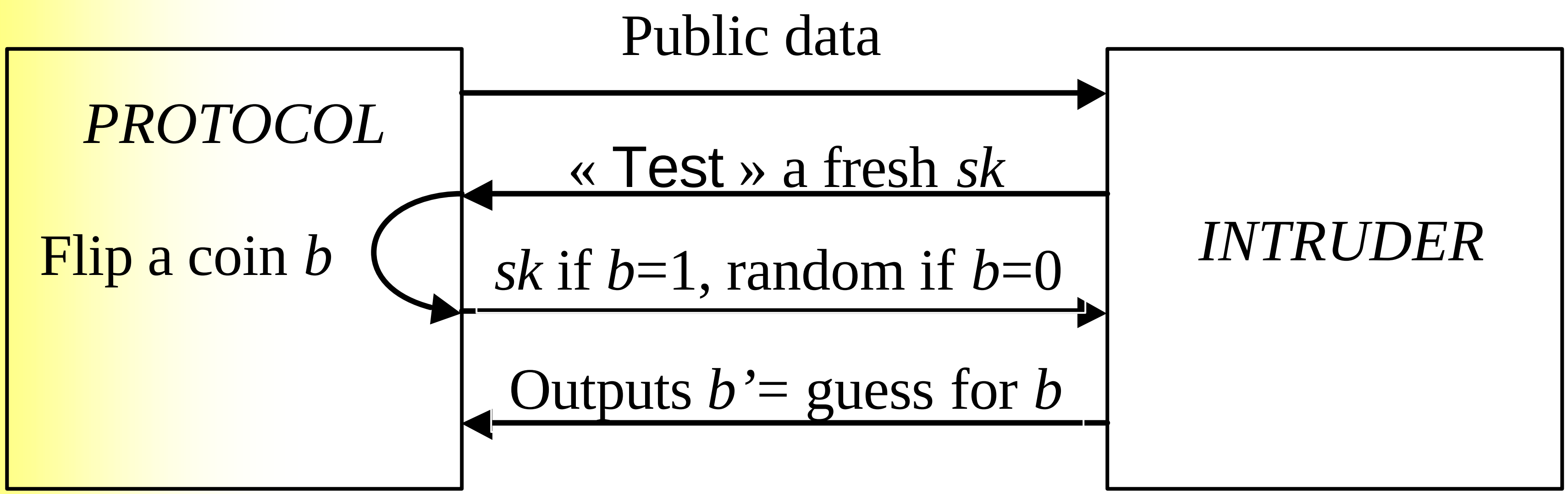
- These two problems can be reduced to the Decisional Diffie-Hellman Problem…

Other Assumptions

- *Existence of Message Authentication Codes*
  MAC's are used to authenticate (sign) the flows between players
  MACs exist if OW-functions exist.

- *Entropy-Smoothing Property*
  The distribution provided by universal hash functions is statistically undistinguishable from a uniform distribution

Security Property



- Security is measured as the adversary's advantage in guessing the bit b involved in the Test-query

Security Theorem

- This advantage is a function of
  - the adversary's advantage in breaking the Group DDH
  - the adversary's advantage in breaking the MAC scheme
  - the adversary's advantage in breaking the Multi-DDH

- Theorem

$$\mathrm{Adv}^{\mathrm{ake}}(T,Q) \;?\; 2nQ\cdot\mathrm{Adv}^{\mathrm{gddh}}(T') + n(n\text{-}1)\cdot\mathrm{Succ}^{\mathrm{cma}}(T)$$
$$+ 2\cdot\mathrm{Adv}^{\mathrm{mddh}}(T') + \text{« negligible terms »}$$

$$T' \;?\; T + nQ\cdot T_{\exp}(k)$$

Discussion

- This theorem has been proved
  - in the presence of concurrent sessions of the protocol
  - in a dynamic context (i.e. together with *Join* and *Leave* protocols in addition to the *Setup* protocol that we presented)
- We also analysed this protocol using a “logical” approach

Discussion (cont.)

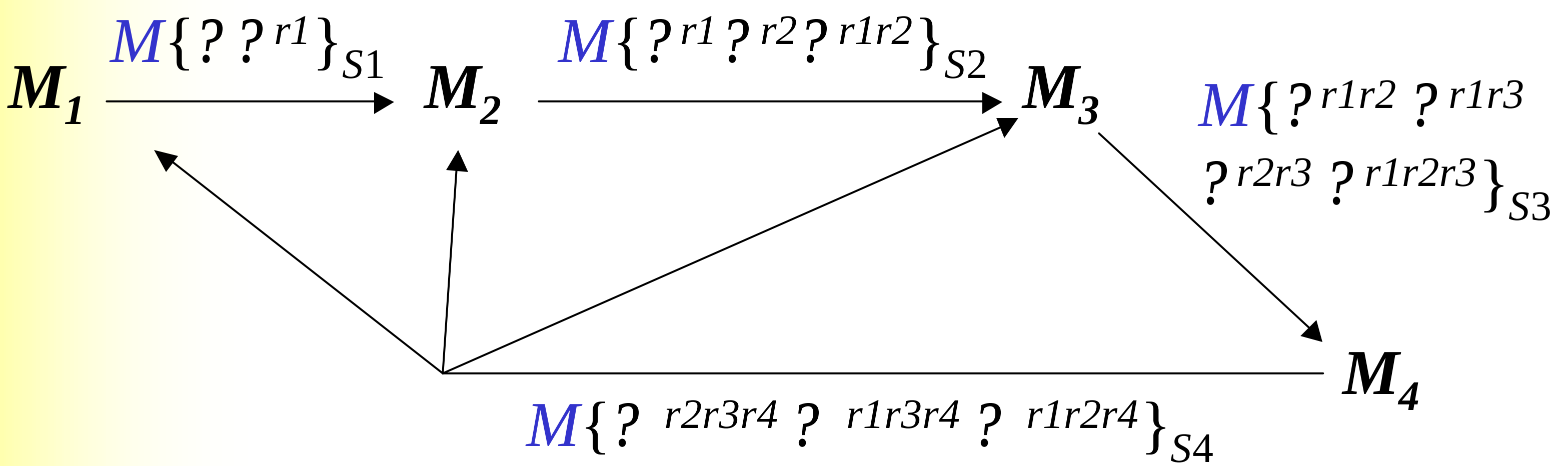
- The computational approach was useful
  - to determine the part of the complexity of the hard problems (Group Decisional Diffie-Hellman, …) injected in the protocol.

In the logical approaches we used, the size of the security parameters is not taken into account…

Discussion (cont.)

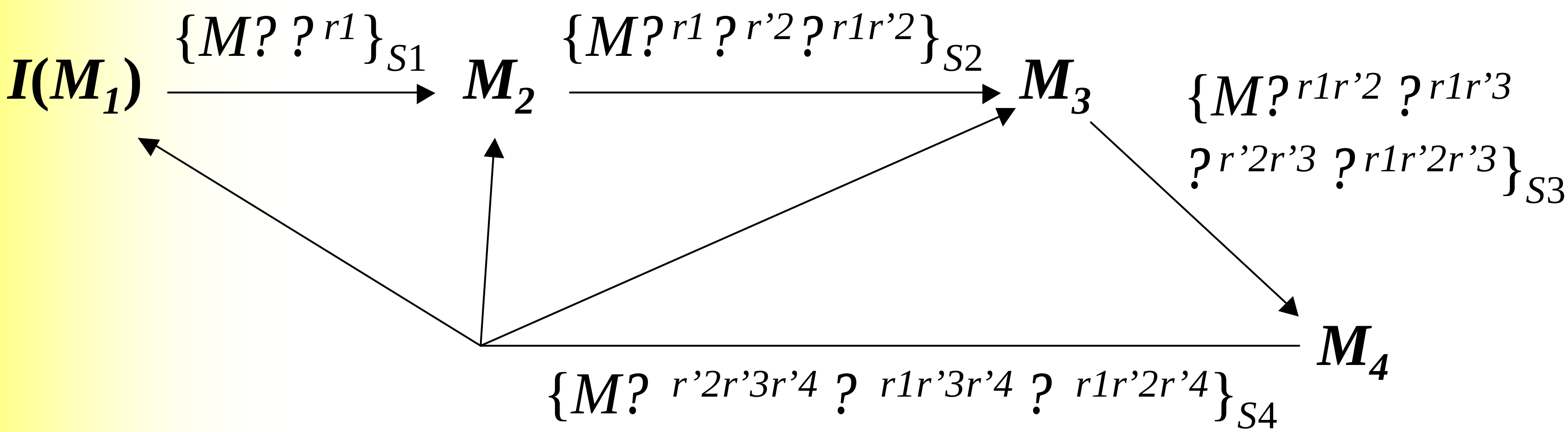
- The logical approach was useful
  - to understand how to construct the messages
  - to understand the causal relations between messages (and so avoid redundancies…)
  - to « measure » the recency of the exchanged terms

Ex: The “computational” security theorem remains correct for the following protocol:



Discussion (cont.)

- *Ex* (*2*): The logical approach is suitable to check freshness properties and the consequences of compromises



If we assume that an old r_1 can be compromised, replay attacks are possible (resulting in new keys compromise…)
Solution to this problem: add nonces or timestamps to identify the sessions…

Conclusion

- Both approaches are providing specific and complementary information…
- First attempts to combine their benefits have been presented:
  - Abadi and Rogaway (2000)
  - Pfitzmann, Schunter and Waider (2000)
  - Guttman, Thayer, Zuck (2001)
- This remains a research in progress…